

Sistem Pengenalan Huruf Braille Menggunakan Metode Deep Learning Berbasis Website

I Made Agus Dwi Suarjaya[#], Bayu Adhya Wiratama[#], Ayu Wirdiani[#]

[#] *Teknologi Informasi, Fakultas Teknik, Universitas Udayana, Denpasar, Indonesia*
E-mail: agussuarjaya@jit.unud.ac.id, bayuadhyawiratama@student.unud.ac.id, ayuwirdiani@junud.ac.id

ABSTRACTS

Braille letters are used as a written language for people with visual impairments. To this day, Braille letters are used in inclusive schools where they are taught to disabled students. However, there are physical capability barriers faced by teachers when correcting Braille answer sheets written by visually impaired students. The ability to read Braille letters is also important for family members to support the students' learning process. This research's purpose was to create a system that can transliterate Braille letters into the Latin alphabet using deep learning methods. The proposed deep learning methods include Base Convolutional Neural Network (CNN), ResNet50, VGG-16, and Inception-v3. The Braille Character image dataset used consists of 12,641 data divided into 37 classes from the AEyeAlliance repository. The Base CNN model used achieved 98% training accuracy, 99% validation accuracy, and 99.1% testing accuracy

Manuscript received June 8, 2024; revised Jul 17, 2024
accepted Jul 30, 2024 Date of publication Sep 30, 2024
International Journal, JITSI : Jurnal Ilmiah Teknologi Sistem Informasi licensed under a Creative Commons Attribution-Share Alike 4.0 International License



ABSTRAK

Huruf Braille digunakan sebagai bahasa tulis bagi penyandang tunanetra. Hingga saat ini, Huruf Braille digunakan pada kegiatan belajar mengajar di sekolah inklusif. Namun, terdapat hambatan kapabilitas fisik yang dialami oleh guru pengajar dalam mengoreksi lembar jawaban siswa tunanetra yang ditulis dalam Huruf Braille. Kemampuan membaca Huruf Braille juga penting dimiliki oleh pihak keluarga untuk membantu kegiatan belajar siswa di rumah. Penelitian ini menghasilkan output berupa sistem yang dapat mentransliterasikan Huruf Braille ke dalam Alfabet Latin menggunakan metode Deep Learning. Metode deep learning yang diajukan antara lain Base Convolutional Neural Network (CNN), VGG-16, ResNet50, dan Inception-v3. Dataset citra Karakter Braille yang digunakan yaitu dataset repositori AEyeAlliance dengan total 12641 data yang dibagi ke dalam 37 kelas. Model Base CNN yang digunakan menghasilkan training accuracy sebesar 98%, validation accuracy sebesar 99%, dan testing accuracy sebesar 99.1%.

Keywords / Kata Kunci — *Visi Komputer, Jaringan Saraf Konvolusi, Pembelajaran Mendalam, Huruf Braille*

CORRESPONDING AUTHOR

Bayu Adhya Wiratama
Teknologi Informasi, Fakultas Teknik, Universitas Udayana, Denpasar, Indonesia
Email: bayuadhyawiratama@gmail.com

1. PENDAHULUAN

Huruf Braille merupakan sistem tulisan (writing) yang digunakan oleh penyandang disabilitas Tunanetra. Huruf Braille dibuat dengan rupa titik-titik timbul dalam pola 2 kolom dan 3 baris dengan masing-masing huruf alfabet memiliki posisi titik yang berbeda-beda [1]. Huruf Braille diciptakan oleh Louis Braille pada abad ke-18.

Huruf Braille digunakan oleh penyandang disabilitas Tunanetra untuk komunikasi tertulis, baik untuk penerimaan maupun penyampaian informasi. Dengan perkembangan jaman, kebutuhan penerimaan informasi penyandang disabilitas Tunanetra mulai digantikan dari Huruf Braille dengan teknologi seperti screen readers. Namun demikian, Huruf Braille masih merupakan fundamental penting untuk dikuasai oleh penyandang disabilitas Tunanetra [2].

Hingga saat ini, Huruf Braille masih digunakan dalam kegiatan belajar mengajar sekolah inklusif, salah satunya adalah SLB 1 Denpasar. Huruf Braille digunakan dalam buku belajar (textbook) dan lembar jawaban siswa. Namun dalam kenyataannya, guru mengalami kendala dalam memeriksa dan mengoreksi lembar jawaban siswa yang ditulis dalam huruf Braille. Berbeda dengan penyandang Tunanetra yang membaca huruf Braille dengan cara diraba, orang normal membaca huruf Braille dengan cara melihat pola dari masing-masing tulisan. Hal ini menyebabkan kelelahan dikarenakan minimnya kontras antara huruf Braille dengan warna kertas yang digunakan. Keberhasilan pendidikan siswa juga tidak luput dari kontribusi orang tua dan keluarga. Namun tidak semua orang memahami cara membaca huruf Braille, sehingga orang tua tidak mampu terlibat dalam mendukung pendidikan anak di rumah.

Optical Braille Recognition (OBR) merupakan jenis penelitian yang bertujuan untuk membuat sebuah sistem yang dapat mengenali huruf Braille menjadi bahasa natural (natural language) yang diinginkan. [3] melakukan ulasan terhadap penelitian mengenai OBR dari masa ke masa. Penelitian tersebut menghasilkan kesimpulan bahwa terdapat beberapa tahap utama dalam melakukan pengenalan huruf Braille, diantaranya Image Acquisition, Image Preprocessing, Braille Dots Detection, dan Braille Cell Construction, diakhiri dengan Braille Recognition. Mengutip dari penelitian [3], Berikut merupakan perbandingan beberapa penelitian pada masing-masing tahap OBR.

Pada tahap image acquisition, berbagai penelitian menggunakan scanner untuk menangkap citra Braille [4]-[6]. Kemudian penelitian [7] dan [8] menggunakan kamera smartphone sebagai device untuk melakukan image acquisition. Hasil yang didapatkan dengan menggunakan scanner lebih baik dibanding menggunakan kamera smartphone, namun kemampuan untuk mengenali huruf Braille dari hasil menggunakan kamera smartphone akan sangat membantu pihak-pihak yang membutuhkan akses terhadap pengenalan huruf Braille. Pada tahap image preprocessing, tujuan yang berusaha dicapai adalah memperbaiki eror yang dihasilkan dari proses image acquisition. Proses utama yang dilakukan antara lain pengubahan ruang warna citra dari RGB menjadi grayscale (grayscale), noise removal, dan de-skewing. Penelitian [9] menerapkan algoritma median filter untuk menghapus noise (noise removal). Kemudian penelitian [10] menerapkan algoritma hough transform dalam melakukan proses de-skewing citra Braille. Tahap Braille Dots Detection bertujuan untuk mendeteksi titik-titik Braille pada citra. Tahap ini dilakukan dengan memisahkan antara titik-titik Braille dengan background yang akan menghasilkan citra biner. Metode yang banyak dipakai oleh penelitian terdahulu adalah thresholding, seperti static thresholding [11] dan local thresholding [10]. Hasil dari thresholding diproses lebih lanjut dengan menggunakan algoritma edge detection menggunakan metode Canny [1] dan Sobel [12]. Digunakan juga operasi morfologi seperti dilation dan erosion [11], [9]. Tahap selanjutnya yaitu Cell Recognition and Transcription yang dilakukan untuk mengelompokkan titik-titik Braille menjadi satu sel yang terdiri dari 3 baris dan 2 kolom. [5] dan [9] menggunakan metode Adaptive Grid Construction yang mempertimbangkan faktor proyeksi horizontal dan vertikal. Masing-masing sel kemudian dikenali dengan menggunakan kode untuk masing-masing karakter atau huruf Braille.

Meski dapat mengenali huruf Braille, namun metode OBR tradisional masih memiliki kesulitan untuk diaplikasikan pada kasus nyata. Proses panjang yang dilakukan juga berpengaruh terhadap kecepatan pengenalan karakter Braille. Tahap pengenalan Braille juga dilakukan dengan menggunakan kode baku yang sangat bergantung pada proses deteksi titik. Hal ini menjadi masalah ketika terdapat titik yang tidak dikelompokkan dengan sempurna pada proses cell recognition (terpotong atau miring). Untuk itu, perlu adanya pembaruan teknologi dalam bidang pengenalan Braille agar dapat menghasilkan sistem yang lebih robust dan efisien.

Penelitian ini mengusulkan proses yang menggunakan Artificial Intelligence, dalam hal ini Deep Learning, untuk mengenali karakter atau huruf Braille yang terkandung dalam sebuah citra. Penelitian ini menggunakan model YOLO (You Only Look Once) [13] untuk secara langsung mendeteksi dan mensegmentasi sel Braille tanpa memerlukan proses preprocessing. Hasil dari proses segmentasi tersebut kemudian dikenali oleh model CNN (Convolutional Neural Network) untuk mendapatkan karakter alfabet dari sel Braille yang dikenali. Penelitian ini juga melakukan komparasi performa model image classification untuk mengetahui model mana yang memiliki performa terbaik dalam mengenali karakter Braille, diantaranya Base CNN, VGG [14], ResNet [15], dan Inception [16]. Proses post-processing dilakukan pada hasil pengenalan dengan model image classification untuk mengubah simbol dan angka. Seluruh sistem pengenalan pada penelitian ini diluncurkan dalam sebuah website agar dapat digunakan oleh user dengan mudah

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan arsitektur YOLOv8 untuk melakukan segmentasi terhadap masing-masing huruf Braille dalam sebuah dokumen. Masing-masing hasil segmentasi dari proses tersebut kemudian akan dikenali oleh

model image classification yang dibangun menggunakan arsitektur convolutional neural network (CNN). Gambar 1 memperlihatkan gambaran besar proses pengenalan citra huruf Braille.



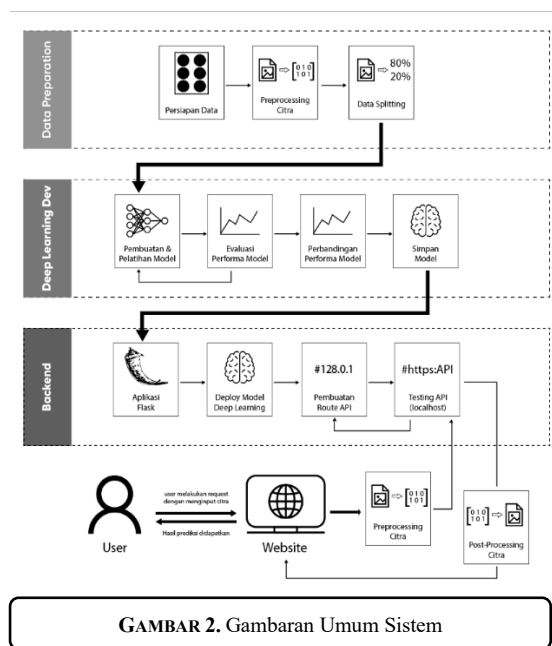
GAMBAR 1. Gambaran Besar Proses Pengelana Citra

Model YOLOv8 yang digunakan untuk segmentasi citra adalah model pretrained bernama DotNeuralNet [17] yang telah dilatih pada berbagai dataset publik diantaranya DSBI [18], AngelinaBraille [19], Natural Braille [20], dan dataset Kaggle. Masing-masing hasil segmentasi kemudian dikenali oleh sebuah model image classification berbasis convolutional neural network (CNN). Model image classification yang digunakan pada penelitian ini merupakan hasil komparasi empat (4) buah model, diantaranya Base CNN, VGG-16 [15], ResNet50 [15], dan Inception-v3 [16].

Dataset yang digunakan untuk melatih model image classification adalah dataset huruf Braille publik yang bersumber dari repositori AEyeAlliance [21]. Dataset tersebut terdiri dari 37 kelas dan 12.000 lebih citra huruf Braille dengan ukuran rata-rata resolusi 90 x 118 pixel. Dataset yang telah dikumpulkan kemudian dilakukan preprocessing agar dapat digunakan pada tahap training model. Preprocessing yang dilakukan adalah resizing nilai piksel agar menjadi rentang 0 hingga 1. Data yang telah diproses kemudian dibagi (split) menjadi data latih (train) dan data validasi (validation) dengan rasio 80:20. Data latih (train) digunakan pada tahap pelatihan model dan data validasi (validation) digunakan pada tahap evaluasi. Setelah masing-masing model dilatih, dilanjutkan pada proses evaluasi untuk mengetahui performa dari masing-masing model. Pada proses ini, digunakan metrik akurasi (1) dan loss (2) dengan rumus berikut [22].

$$accuracy = \frac{tp+tn}{tp+tn+fn+fp} \quad (1)$$

$$loss = y \times \log(p) + (1 - y) \times \log(1 - p) \quad (2)$$



GAMBAR 2. Gambaran Umum Sistem

Model yang memiliki performa tertinggi akan digunakan pada saat proses inferens atau prediksi. Seluruh proses tersebut diluncurkan dalam bentuk website agar dapat dengan mudah digunakan oleh user. Website dibangun menggunakan bantuan framework web development Python yaitu Flask. Gambar 2 memperlihatkan gambaran umum pembuatan sistem

Citra yang diinput oleh user akan masuk pada proses preprocessing citra. Proses ini melakukan segmentasi terhadap masing-masing huruf Braille dalam citra input. Hasil segmentasi kemudian akan diresize sesuai ukuran input model image classification CNN untuk kemudian dikenali. Hasil pengenalan tersebut akan diproses kembali (postprocessing) seperti pengubahan simbol dan angka, pengenalan spasi, dan penghapusan overlapping bounding box. Hasil pengenalan yang telah diproses kemudian ditampilkan di atas citra input. Citra hasil pengenalan tersebut kemudian ditampilkan pada website.

3. HASIL DAN PEMBAHASAN

3.1. Hasil

Masing-masing model dilatih menggunakan dataset dan *hyperparameter* yang sama yaitu 50 *epoch* dan *batch size* 64. Performa masing-masing model *image classification* dibandingkan menggunakan metrik akurasi dan *loss*. Tabel 1 memperlihatkan performa dari masing-masing model. Kedua metrik tersebut kemudian diplot pada *learning curve* [23] untuk melihat apakah model mengalami *overfitting* atau *underfitting*. Gambar 3 menunjukkan *learning curve* dari masing-masing model.

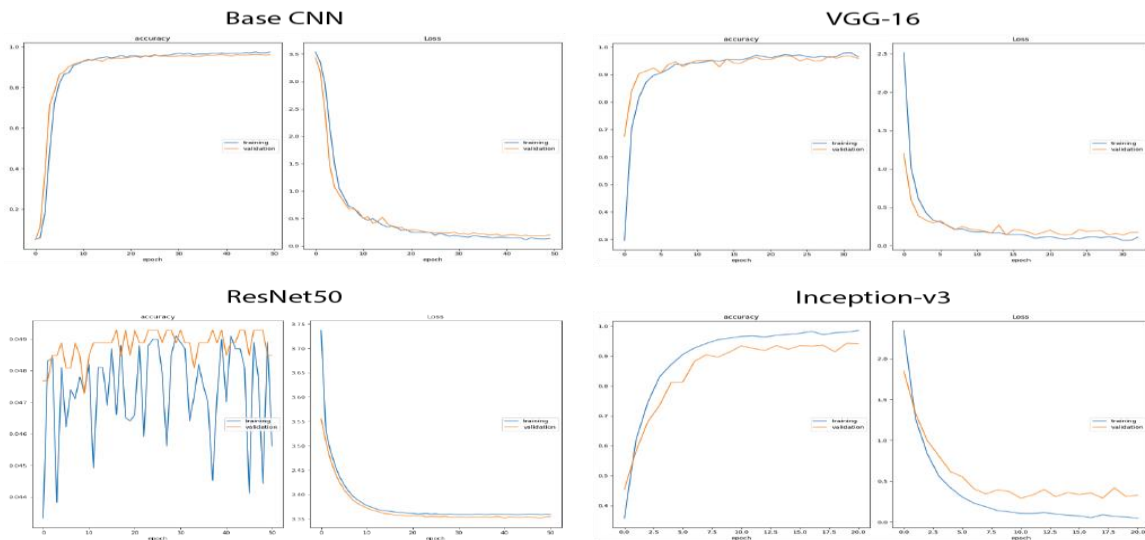
TABEL 1. Hasil Evaluasi Performa Model Image Classification

Metriks	Model			
	CNN	VGG-16	ResNet50	Inceptionv3
Akurasi (Training)	98%	97%	61%	95%
Akurasi (Validasi)	99%	98%	68%	95.5%
Loss (Training)	0.06	0.082	1.203	0.149
Loss (Validasi)	0.07	0.118	1.189	0.253

Model dengan performa tertinggi (*Base CNN*) kemudian digunakan pada tahap inferens atau prediksi. Data uji yang digunakan merupakan 14 lembar soal ujian TIK yang didapat dari SLB 1 Denpasar dengan rata-rata 400 huruf Braille dalam satu dokumen. Tabel 2 berikut menunjukkan hasil pengujian model pada data uji.

TABEL 2. Hasil Pengujian Model pada Data Uji

Citra Uji	Jumlah Karakter Braille dalam Citra	Jumlah Karakter yang Terdeteksi YOLO	Jumlah Karakter yang Benar Dikenali CNN
Braille-1.png	478	464 (97%)	460 (99.1%)
Braille-2.png	356	345 (96.9%)	343 (99.4%)
Braille-3.png	402	387 (96.2%)	383 (98.9%)
Braille-4.png	399	389 (97.4%)	383 (98.4%)
Braille-5.png	429	412 (96%)	410 (99.5%)
Braille-6.png	400	386 (96.5%)	385 (99.7%)
Braille-7.png	382	364 (95.2%)	359 (98.6%)
Braille-8.png	411	406 (98.7%)	400 (98.5%)
Braille-9.png	451	442 (98%)	436 (98.6%)
Braille-10.png	343	334 (97.3%)	330 (98.8%)
Braille-12.png	453	446 (98.4%)	441 (98.8%)
Braille-13.png	393	387 (98.4%)	384 (99.2%)
Braille-14.png	343	330 (96.2%)	330 (100%)
Braille-15.png	29	29 (100%)	29 (100%)
RATA-RATA		97.3%	99.1%

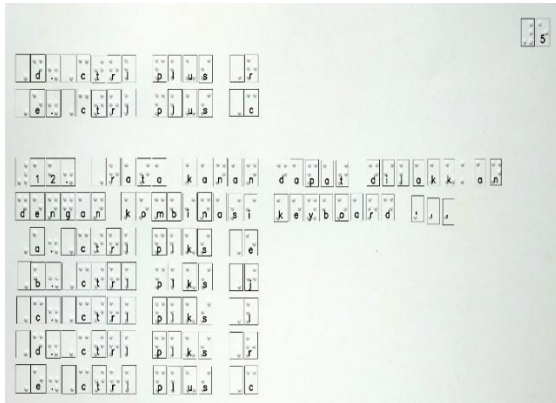


GAMBAR 3. Grafik Evaluasi Performa Model Image Classification

Tabel 3 merupakan perbandingan antara hasil penelitian ini dengan hasil dari beberapa penelitian yang diulas oleh [3].

TABEL 3. Hasil Pengujian Model pada Data Uji

Paper	Teknik Dots Detection	Teknik Cell Recognition	Teknik Transcription	Hasil	Kecepatan/Halaman
[25]	Dynamic Local Thresholding	Placement of Braille Dots	Binary Digital with lookup table	N/A	2 detik
[26]	Cross-correlation approach for dots detection	Multilayer Perceptron Neural Network	Binary Digital with lookup table	98%	N/A
[10]	Local dynamic thresholding to separate dots into three classes	placement of Braille dots	Binary digital with 6 bits	95%	N/A
[18]	Binary module by taking a window to crop from the image a sub image the size of a Braille dot SVM classifier to detect dots	Placement of Braille dots	searching algorithm and a look-up table	95%	N/A
[5]	Calculate threshold using image histogram	Using horizontal and vertical projection profiling	Binary vector	98.9%	N/A
[11]	Filling the Braille dots Edge detection using canny edge detector	Placement of braille dots	Decimal code with lookup table	N/A	32.6 detik menggunakan Matlab
Penelitian ini	YOLOv8		Convolutional Neural Network (CNN)	99.1%	21 detik

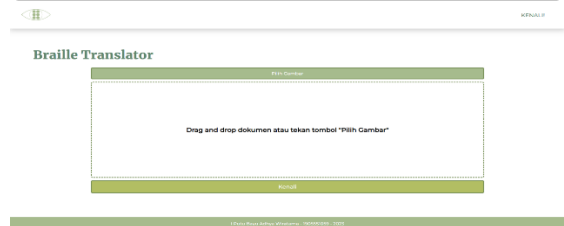


GAMBAR 4. Hasil Pengenalan pada Data Uji

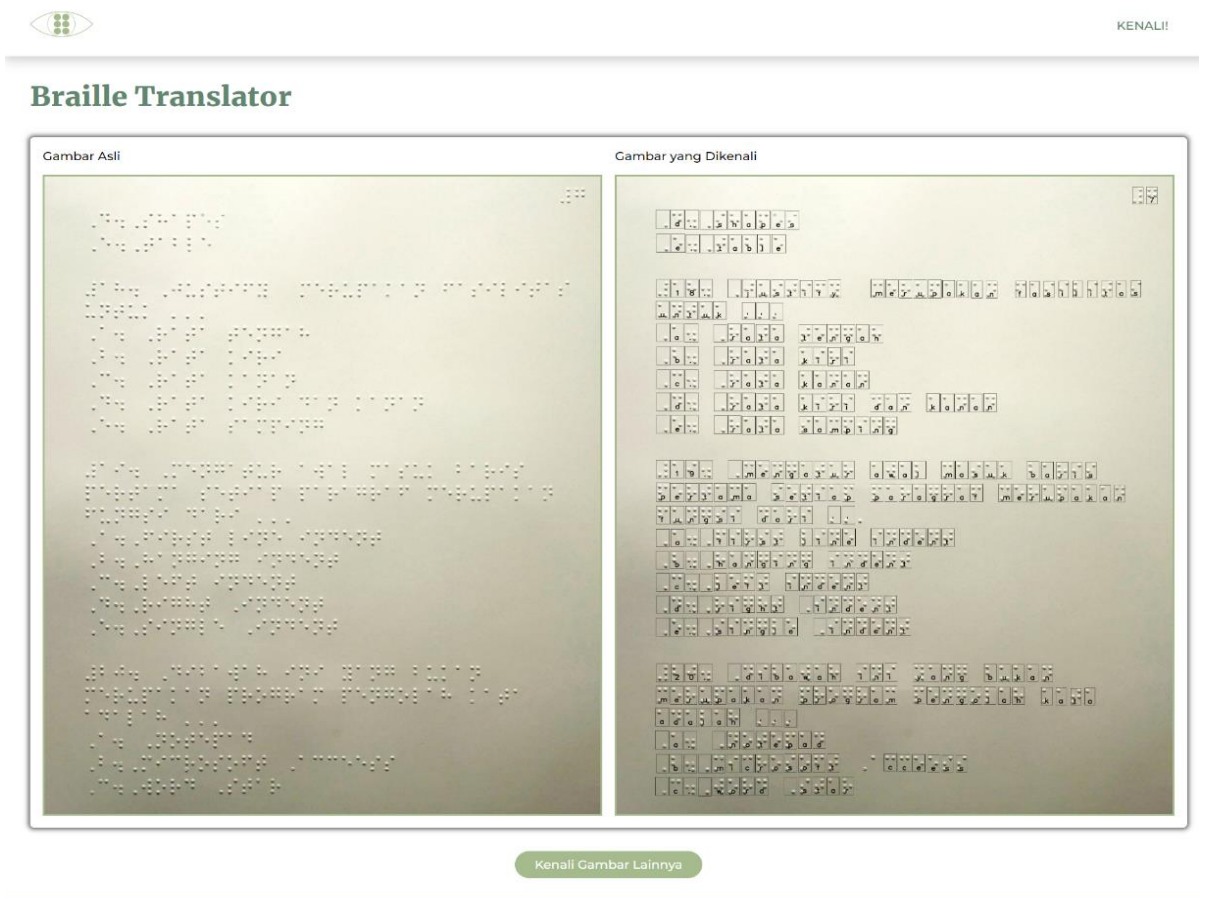
Seluruh alur pengenalan yang telah dibangun diluncurkan (*deploy*) dalam bentuk *website* agar dapat memudahkan *user* dalam menggunakan *service* sistem. *Website* dibangun menggunakan *framework* Flask, bahasa pemrograman Python, serta HTML dan CSS. Gambar 4 berikut menunjukkan hasil pengenalan model pada sebuah data uji berisi sebagian soal TIK yang ditulis dalam huruf Braille. Gambar 5 hingga Gambar 7 menunjukkan tampilan *website*.



GAMBAR 5. Halaman *Landing Page*



GAMBAR 6. Halaman *Upload File*



Kenali Gambar Lainnya

I Putu Bayu Adhya Wiratama - 1905551059 - 2023

GAMBAR 7. Halaman Hasil Pengenalan

3.2. Pembahasan

Model *Base CNN* menghasilkan skor 98% pada akurasi pelatihan (*training*), 99% pada akurasi validasi (*validation*), 0.06 pada *loss* pelatihan (*training*), dan 0.07 pada *loss* validasi, serta mengalami *goodfit* seperti yang dapat dilihat pada Gambar 3 (*Base CNN*). Model VGG-16 menghasilkan skor 97% pada akurasi pelatihan (*training*), 98% pada akurasi validasi (*validation*), 0.082 pada *loss* pelatihan (*training*), dan 0.118 pada *loss* validasi (*validation*), serta mengalami *goodfit* seperti yang dapat dilihat pada Gambar 3 (VGG-16). Model ResNet50 menghasilkan skor 61% pada akurasi pelatihan (*training*), 68% pada akurasi validasi (*validation*), 1.203 pada *loss* pelatihan (*training*), dan 1.189 pada *loss* validasi (*validation*), serta mengalami *underfitting* dikarenakan model tidak menunjukkan adanya generalisasi baik itu pada data training maupun validasi [24]. Model Inception-v3 menghasilkan skor 95% pada akurasi pelatihan (*training*), 95.5% pada akurasi validasi (*validation*), 0.149 pada *loss* pelatihan (*training*), 0.253 pada *loss* validasi (*validation*), serta mengalami sedikit *overfitting* dikarenakan model tidak menunjukkan adanya generalisasi pada data validasi. Model *Base CNN* digunakan sebagai model inferens pada tahap pengujian (*testing*) karena menghasilkan performa tertinggi.

Pada tahap pengujian, model *pretrained YOLOv8* digunakan untuk mendeteksi dan melakukan segmentasi terhadap masing-masing huruf Braille yang ditemukan pada data uji. Rata-rata akurasi YOLOv8 dalam mengenali seluruh huruf Braille dengan benar adalah sebesar 97.3% seperti pada Tabel 2. Hasil dari proses segmentasi oleh YOLOv8 kemudian dikenali oleh model *Base CNN* dengan rata-rata akurasi mencapai 99.1%. Hasil ini menunjukkan bahwa proses dan sistem yang dibuat mampu digunakan untuk mengenali huruf Braille dalam sebuah citra dokumen Braille.

4. KESIMPULAN

Sistem pengenalan huruf Braille dibuat untuk membantu seluruh pihak yang membutuhkan kemampuan untuk mengenali huruf Braille, baik itu guru sekolah inklusif, orang tua siswa Tunanetra, maupun pihak lainnya. Penelitian ini menghasilkan sebuah *website* yang dapat mengenali huruf Braille menggunakan bantuan *deep learning*, spesifik pada teknologi *computer vision* menggunakan YOLOv8 dan CNN. Sistem ini dapat diteliti kembali baik dari segi pengembangan *software* maupun penggunaan metode yang berbeda.

REFERENSI

- [1] Kavalgeri, S. A., Chakraborty, R. V., Naz, S. F., & Chaitanya, K. J. (2019). E-Braille: A Study Aid for Visual Impaired. *Ijresm*, 2(April), 131–135. Bereiter-Hahn, J. (1990). Behavior of mitochondria in the living cell. *International review of cytology*, 122, 1-63
- [2] Guerreiro, J., Gonçalves, D., Marques, D., Guerreiro, T., Nicolau, H., & Montague, K. (2013). The today and tomorrow of Braille learning. *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2013, May 2014*, 20–22. <https://doi.org/10.1145/2513383.2513415> Dallas, C., Gerbi, A., Tenca, G., Juchaux, F., & Bernard, F. X. (2008). Lipolytic effect of a polyphenolic citrus dry extract of red orange, grapefruit, orange (SINETROL) in human body fat adipocytes. Mechanism of action by inhibition of cAMP-phosphodiesterase (PDE). *Phytomedicine*, 15(10), 783-792
- [3] Isayed, S., & Tahboub, R. (2015, August 19). A review of optical Braille recognition. 2015 2nd World Symposium on Web Applications and Networking, WSWAN 2015. <https://doi.org/10.1109/WSWAN.2015.7210343> Garnier, A., Fortin, D., Zoll, J., N'Guessan, B., Mettauer, B., Lampert, E., Veksler V., & Ventura-Clapier, R. (2005). Coordinated changes in mitochondrial function and biogenesis in healthy and diseased human skeletal muscle. *The FASEB Journal*, 19(1), 43-52
- [4] Hasnat, A., & Nasipuri, M. (2013). Development of a Bangla Character Recognition (BCR) System for Generation of Bengali Text from Braille Notation. In *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* (Issue 3).
- [5] Padmavathi, S., K.S.S, M., Reddy .S, S., & D, Meenakshy. (2013). Conversion of Braille to Text in English, Hindi and Tamil Languages. *International Journal of Computer Science, Engineering and Applications*, 3(3), 19–32.
- [6] T, Shreekanth., & Udayashankara, V. (2013). A Review on Software Algorithms for Optical Recognition of Embossed Braille Characters. *International Journal of Computer Applications*, 81(3), 25–35.
- [7] Murray, I., & Pasquale, A. (2006). A portable device for the translation of braille to text. Eighth International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2006, 2006, 231–232.

- [8] Venugopal-Wairagade, G. A. (2016). Braille Recognition using a Camera-enabled Smartphone. *International Journal of Engineering and Manufacturing*, 6(4), 32–39.
- [9] Scholar, R. (2014). An Application of Eight Connectivity based Two-pass Connected-Component Labelling Algorithm For Double Sided Braille Dot Recognition. In *International Journal of Image Processing (IJIP)* (Issue 8).
- [10] Antonacopoulos, A., & Bridson, D. (2004). A robust braille recognition system. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3163, 533–545.
- [11] Khalil, S., Al-Shamma, S. D., & Fathi, S. (2010). Arabic Braille Recognition and Transcription into Text and Voice The Applications of Tele-diagnostic for Disease Detection and Management Project View project Arabic Braille Recognition and Transcription into Text and Voice.
- [12] Ng, C. M., Ng, V., & Lau, Y. (1999). Regular feature extraction for recognition of Braille. *Proceedings - 3rd International Conference on Computational Intelligence and Multimedia Applications, ICCIMA 1999*, 302–306.
- [13] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection.
- [14] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition.
- [15] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 770–778.
- [16] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 2818–2826.
- [17] Github.com. (2023, 28 Maret). DotNeuralNet. Diakses pada 15 Juni 2023, dari <https://github.com/snoop2head/DotNeuralNet>.
- [18] Li, R., Liu, H., Wang, X., & Qian, Y. (2018). DSBI: Double-Sided Braille image dataset and algorithm evaluation for Braille dots detection. *ACM International Conference Proceeding Series*, January, 65–69.
- [19] Ovodov, I. G. (2021). Optical Braille Recognition Using Object Detection Neural Network. *Proceedings of the IEEE International Conference on Computer Vision, 2021-Octob*, 1741–1748.
- [20] Lu, L., Wu, D., Xiong, J., Liang, Z., & Huang, F. (2022). Anchor-Free Braille Character Detection Based on Edge Feature in Natural Scene Images. *Computational Intelligence and Neuroscience*, 2022.
- [21] Github.com. (2018, 9 September). Aeye-alliance. Diakses pada 15 Juni 2023, dari <https://github.com/HelenGezahegn/aeeye-alliance>.
- [22] Hidra Amnur, Y. Syanurdi, R. Idmayanti, and A. Erianda, "Developing Online Learning Applications for People with Hearing Impairment," vol. 5, no. March, pp. 32–38, 2021, [Online]. Available: www.joiv.org/index.php/joiv
- [23] M, H., & M.N, S. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2), 01–11.
- [24] Viering, T., & Loog, M. (2021). The Shape of Learning Curves : a Review. 1–20.
- [25] Ghasemian, A., Hosseinmardi, H., & Clauset, A. (2020). Evaluating overfit and underfit in models of network community structure. *IEEE Transactions on Knowledge and Data Engineering*, 32(9), 1722–1735.
- [26] Zhang, S., & Yoshino, K. (2007). A Braille Recognition System by the Mobile Phone with Embedded Camera. *Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007)*, 223-223.
- [27] Morgavi, G., and Morando. M. "A neural network hybrid model for an optical braille recognizer." *International Conference on Signal, Speech and Image Processing 2002 (ICOSSIP 2002)*. 2002.